Before the
Federal Communications Commission
Washington DC 20554

In the Matter of	)	
	)	
Authorization and Use of Software	)	ET Docket No. 00-47
Defined Radios	)	

COMMENTS OF VANU, INC.

Mitchell Lazarus
FLETCHER, HEALD & HILDRETH, P.L.C.
1300 North 17th Street, 11th Floor
Arlington, VA 22209
703-812-0440

March 19, 2001

TABLE OF CONTENTS

Before the
Federal Communications Commission
Washington DC 20554

In the Matter of	)	
	)	
Authorization and Use of Software Defined Radios	)	ET Docket No. 00-47
	)	

COMMENTS OF VANU, INC.

Vanu, Inc. hereby files these comments in the above-captioned proceeding .[1]

INTRODUCTION

A. Summary

Vanu proposes a small but critical change in the proposed rules for certifying software-defined radios (SDRs). Specifically, we suggest an alternative to the requirement for separate testing of every hardware/software combination.

"Platform software" and "signal-processing software." Vanu proposes a distinction between "platform software," which manages critical transmitter functions such as frequency, power, and bandwidth; and "signal-processing software," which embodies the details of the particular modulation in use.

The platform software will present a standard interface, so the same signal-processing software can run on several hardware platforms. This greatly reduces the cost of both layers. The cost of signal-processing software can be amortized over many platforms, and the cost of platform software, over many radio services.

Testing rule. Applicants should be required to test the hardware together with the platform software, but should be permitted to test the signal-processing software separately. This

---

[1] *Authorization and Use of Software Defined Radios*, ET Docket No. 00-47, Notice of Proposed Rule Making, FCC 00-430 (released Dec. 8, 2000) (Notice).

scheme will completely address all of the Commission's concerns about radio performance, yet will reduce costs and facilitate innovation.

Interference prevention; RF safety. The platform software manages the hardware under control of the signal-processing software. The platform responds to requests from the signal-processing software, much as a PC operating system and its device drivers respond to hardware requests from application software. The platform software monitors the behavior of the signal-processing software, and turns off the radio if it goes out of compliance.

As an added precaution, the signal-processing software carries an encrypted software tag that specifies the limits set out in the corresponding technical rules. The platform software will reject transmission requests that exceed those limits.

Software reliability. *Separate certification of platform software and signal-processing software will greatly reduce the likelihood of harmful software errors.* There are two reasons. First, the platform software has a much simpler logic flow than the signal-processing software (or the monolithic software contemplated by the Notice). The smaller number of branches, paths, and states makes the platform software more amenable to thorough testing, and hence results in extremely high reliability. Second, because the platform software will support a wide variety of signal-processing programs over several generations, it can justify a much greater investment in reliability testing. Even if the signal-processing software carries hidden defects, carefully debugged platform software makes it extremely unlikely that those defects can take the radio out of compliance.

B. About Vanu, Inc.

Vanu was formed in 1998 to explore the feasibility of building software radios using object oriented computer languages running on general purpose processors. The extent to which Vanu uses software to implement signal processing distinguishes us from other radio developers. In the nomenclature of the SDR Forum, Vanu develops "software radios" as opposed to "software defined radios." Pushing the digitization closer to the antenna permits our products much greater flexibility to adapt the nature of the signal processing performed by the radio.

For more information, see Appendix A and www.vanu.com.

DISCUSSION

C. Software-Defined Radios Will Speed Innovation and Improve Spectrum Efficiency.

Most hardware-based radios can operate in only one or two bands. Historically, this limitation produced the Commission's basic licensing scheme, which restricts each user to specific bands. Even though most licensees do not fill their bandwidth, the Commission's rules generally prohibit a licensee from making its excess capacity easily available to others. As a result, large amounts of spectrum sit idle for much of the time. The much-discussed spectrum shortage is not so much a lack of vacant spectrum as it is a shortage of unlicensed channels.

The Commission is presently investigating mechanisms to make better use of licensed but underused spectrum.[2] Many of the anticipated mechanisms, especially for short-term spectrum transfers, will depend on radios that can be quickly adjusted as to frequency band, modulation,

---

[2] *Promoting Efficient Use of Spectrum Through Elimination of Barriers to the Development of Secondary Markets*, WT Docket No. 00-230, Notice of Proposed Rulemaking, FCC 00-402 (released Nov. 27, 2000).

bandwidth, and power, preferably by remote control from a central location. A true "spot market" in spectrum will require radios capable of being reconfigured on the fly.

These devices will use spectrum more efficiently, and can incorporate new advances in wireless technology very quickly. By contrast, for example, the Commission's modest effort toward improving spectrum efficiency for the Private Land Mobile Radio service is taking decades to complete.[3] When narrower bandwidths and other improvements can be implemented by a simple software upgrade, downloadable over the air, spectrally efficient technologies can be adopted in hours instead of years.

SDRs will also end the need for lock-step, permanent spectrum allocations. Today allocations are national in scope, and many are international, in part because hardware-based radios force a need for geographical uniformity. This practice further harms efficiency. Needs are not uniform everywhere, and can change over time. But SDRs will accommodate allocations that vary from place to place, and from time to time, according to local requirements. In particular, SDRs will permit transparent global roaming, by switching among frequency bands and modulations as needed to take advantage of available services anywhere in the world.

In short, the advent of software-based, wireless infrastructure that can be upgraded over the air will permit fast, low-cost improvements to wireless communications. These have the potential to improve spectrum efficiency, quality of service, diversity of service offerings, and allocation of spectral resources.

---

[3] *See generally* PR Docket No. 92-235 (Private Land Mobile "refarming").

D. Applicants Should Have the Option of Testing Platform Software with the Hardware, and Signal-processing Software Separately.

Vanu agrees in principle with the Commission that hardware and software should be approved together. At this point in development of the technology, it is not possible to predict the relevant RF characteristics of the radio from either the hardware or the software alone.[4]

Nevertheless, only certain software layers are critical to RF operation, and so need be tested together with the hardware.

1. Platform software and signal-processing software

Vanu proposes a regulatory distinction between "platform software" and "signal-processing software," characterized as follows:

- Platform software provides an interface with the hardware, and manages aspects of the device that do not change with the details of the particular modulation used. By design, these include power limits, occupied bandwidth, and out-of-band emissions, as well as system functions such as start-up and shut-down, basic configuration and testing of the hardware devices, and error logging and field test functions.

- Signal-processing software implements the signal processing and state machines that control the underlying platform's communications according to a particular standard (*e.g.*, IS-91 or IS-95). More advanced systems will incorporate routing, a reliable transport layer, and eventually cognitive radio functionality in this layer.

In some respects, this distinction mirrors that between a PC's operating system (such as Windows) and the application software it is executing (such as MS Word).[5] A properly written application program never communicates with the hardware directly, but places all hardware

---

[4] Notice at para. 18.

[5] The PC example is only an analogy, and it overlooks important distinctions. Computer cognoscenti might consider the term "operating system," as used here, to include the PC BIOS, driver software, and controller ROMs.

requests through the operating system. For example, when the user presses "save" in MS Word, to save a file to disk, MS Word does not write the file itself. Instead, it passes the request to Windows, which manages the details of sending the file to the disk drive. This separation of operating system from application functions has several advantages:

(a) The operating system functions need only be written and debugged once, after which any number of application programs can share them.

(b) Writing and upgrading application software is faster and less expensive than otherwise, because even complicated hardware operations can be handled with simple calls to the operating system.

(c) The existence of widely-used operating systems makes it possible to run a single application program on a large variety of makes and models of PCs.

(d) The operating system protects critical functions of the PC. Even a badly written application program cannot damage the hardware, the operating system, or attached networks or peripherals, so long as it conforms to certain basic conventions laid down by the operating system.[6]

All of these advantages apply to the proposed distinction between SDR platform software and signal-processing software. The relatively permanent platform software will directly control the hardware functions that affect interference and RF safety. Initially, at least, only the radio manufacturer should be permitted to change the platform software. But flexible signal-processing software will enable the device owner, service provider, software vendor, or device manufacturer to change the functionality of a device through simple, over-the-air downloads, with no threat to system integrity.

---

[6] Because earlier versions of Windows did not include mechanisms to fully isolate the operating system from applications, the whole system could be corrupted by a misbehaving application. Isolation mechanisms are now widely used, including in Windows NT, leading to highly reliable systems. See Part D.6, below.

2. Basic certification rules

Applicants should have two options for certification: (a) test the hardware together with the complete software package, as contemplated in the Notice, or (b) test the hardware together with the platform software, and test the signal-processing software separately. See also Part D.7, below, on testing details.

The Commission tentatively concluded that testing each hardware/software combination in an SDR would be no more burdensome than the current process, which requires testing each mode in which a radio operates.[7] Vanu respectfully disagrees. *N* current radio platforms, each operating in two or three modes, require only 2*N* or 3*N* separate tests. As the SDR industry develops, however, we foresee large numbers of software packages for multiple hardware platforms. *N* different hardware platforms, each running *P* possible software packages, will require a total of *N* x *P* distinct tests. If the SDR concept is successful, the number of combinations requiring separate testing will become intractable.

As we show below, however, testing the complete software package with every type of hardware is unnecessary. The industry can completely address all of the Commission's concerns about radio performance by testing the hardware only with the platform software. Certifying the signal-processing software separately adds no risks, yet both reduces costs and facilitates innovation.

3. Public interest considerations

We expect manufacturers to establish a standard interface between platform software and signal-processing software. This will enable each type of hardware to support any compatible

---

[7] *See* Notice at para. 18.

instance of signal-processing software, in much the same way that a computer operating system such as Windows or UNIX allows any compatible software to run on any hardware that supports the operating system. Radio manufacturers will have a strong incentive to reach agreement on these standards, because porting signal-processing software to multiple hardware platforms will help to amortize development costs. Development for multiple platforms would be impractical without standardized platform software to reduce the impact of hardware variations.

Prudence favors designing the platform software to be independent of the signal-processing software to the greatest extent possible. This minimizes the engineering cost and effort both of adding a new waveform to a platform, and of developing a new hardware platform for existing modulations. As we discuss in more detail below, isolating the layers also facilitates thorough testing of the platform software, and improves reliability by amortizing the costs of reliability over a larger number of waveforms and applications.

In short, the platform/signal-processing architecture will allow more rapid adoption of new technologies, reduce costs for adopting new technologies, increase the useful life of mobile and fixed hardware, and provide for more reliable software. End users will benefit from faster implementation of new services, greater interoperability, and lower costs for both equipment and services.

The Commission need not mandate this approach. If the rules merely allow separate testing of platform and signal-processing software, we are confident the advantages of separate software layers will lead to widespread industry adoption.

4. Interference prevention; RF safety

In the Commission's view, testing the hardware and software together is the only way to ensure that equipment complies with the technical rules so as to prevent interference or excessive RF radiation.[8] The Commission's purposes can be fully accomplished -- indeed, better accomplished -- by certifying the hardware and *platform* software together. Certifying the signal-processing software separately permits greater protection against interference and RF hazards.

As noted above, the platform software controls all of the transmitter characteristics that most directly affect interference and RF safety, including output power, occupied bandwidth, and out-of-band emissions. The signal-processing software specifies these properties, according to the technical rules of the service in which the radio is operating, and communicates them to the platform.

Three distinct mechanisms ensure that the hardware stays within limits. First, the signal-processing software requests the platform to generate signals that conform to the technical rules for that service. Second, the platform software monitors the behavior of the signal-processing software, and halts the transmission if the limits are exceeded. Third, as an independent back-up, the signal-processing software carries an encrypted software tag that separately specifies the limits set out in the corresponding technical rules. (This information may be part of the authorization tag that certifies authenticity of the signal-processing software.[9]) The platform

---

[8] Notice at para. 18.

[9] *See* Notice at paras. 30-31 (need for software authentication). The same tag encryption that protects the radio against unauthorized software can also protect the operating parameters against unauthorized modification.

software rejects transmission requests that exceed the limits in the tag. Thus, even if the signal-processing software is otherwise defective, the tag still keeps the radio's operation within limits.

This last function is similar to that executed by device drivers in an operating system. Device drivers perform extensive checks on configuration requests by applications. Managing hardware devices that enforce the limits is a natural extension of the platform software's role in managing all the hardware of the system.

5. Software reliability

The Commission asks if it should impose requirements that prevent grantees from "*knowingly* marketing software" that could take an SDR out of compliance with the Commission's rules.[10] We think the answer has to be yes. Equally important, however, is preventing the distribution of software that *inadvertently* takes the radio out of compliance.

All computer users know that software often has bugs. Even software produced by careful engineering teams for use in critical applications is rarely defect-free. ***Certifying platform software and signal-processing software separately will greatly reduce the likelihood of harmful software errors.*** As we show below, this argument is so strong that the Commission may eventually impose the separation of software into platform and signal-processing layers.[11]

Placing the hardware under the control of platform software, controlled in turn by signal-processing software, will minimize the incidence of harmful defects for several reasons.

---

[10] Notice at para. 34 (emphasis added; footnote omitted).

[11] We do not advocate requiring a separation now, because it entails certain drawbacks, such as greater power dissipation due to increased processing and memory requirements. These will diminish with improvements in chip design.

(a) Simplicity aids reliability.

The difficulty in debugging software increases sharply with the number of decision points -- what programmers call "conditional branches" -- and with the number of possible "states" of the software. Each branch doubles the number of possible paths through the program and adds to the possible states. Just thirty branches yield over a billion distinct paths. Typical desktop software has hundreds or thousands of branches, yielding astronomical numbers of paths and states -- far too many to test in a lifetime.

SDR signal-processing software will eventually have to support cognitive radio functionality, spectrum sharing, performance optimization under varying weather, noise, or multipath conditions, and many other features. Taken together, these will require very large numbers of decision points. As the numbers climb, software defects take on subtle and non-linear effects, while the combinatorial explosion in the numbers of paths and states makes the defects harder to detect and locate. Because the monolithic software contemplated in the Notice will incorporate all of the signal-processing software functionality, it will necessarily suffer the same liabilities.

In contrast, the software in current communications devices is reliable because it has relatively few branches and states, and so can be tested thoroughly. The platform software proposed for SDRs, stripped of the signal-processing software complexities, will similarly be small and simple enough for thorough testing and extremely high reliability.

(b) Re-usability justifies greater investment.

Platform software can justify much greater expenditures on reliability than signal-processing software. Platform software will support a wide variety of signal-processing

programs over several generations. Its relatively long market life can recover a substantial investment in development, particularly in reliability testing.

With adequate resources, it is possible to achieve very high confidence in the correct operation of software, even when the paths and decision points are too numerous to be fully tested. Even if the signal-processing software carries hidden defects, carefully debugged platform software makes it extremely unlikely that those defects can take the radio out of compliance.

(c) Software separation reduces the impact of a single error.

If a software fault remains in certified signal-processing software, it is extremely unlikely to cause unsafe transmission, because the platform software will detect the problem and shut off the transmission. Conversely, a software fault in certified platform software is also extremely unlikely to cause unsafe transmission, because the platform software does not contain the code segments that command the hardware to perform transmissions. A bug in the platform software can lead to unsafe transmissions only if the signal-processing software has a precisely corresponding error. Because the platform and signal-processing software will be developed and tested independently, the chance of a common-mode bug will be extremely low.

Completely defect-free platform software (or any non-trivial software) cannot be assured with current software engineering methods. Separating the signal-processing software and the platform software provides protection against unsafe transmissions, even allowing for the remote possibility of faults in platform software.

6. Enforcing separation

Because ultimate control of the hardware rests in the platform software, code and data values in the platform layer must be protected against accidental change by the signal-processing software. Several mechanisms are available to enforce the separation. The platform software can function as a true operating system, using virtual memory and privilege level hardware support in the processor to confine signal-processing software to an address space in which it cannot directly access either the platform software or the hardware. Alternatively, the platform software can run the signal-processing software in a virtual machine, as in the Java language system. Or, the platform software can execute on a different processor and memory subsystem than the signal-processing software. Other, more sophisticated mechanisms are also available.

None of these measures is available in the monolithic, non-layered software contemplated in the Notice. That architecture makes it much more difficult to eliminate the risk of higher-level functions inadvertently modifying lower-level controls.

7. Testing details

(a) Hardware and platform software

In testing each combination of hardware and platform software, the applicant should be required to demonstrate these properties:

(i) At start-up of a waveform, the platform software prevents the execution of unauthorized software, and reads out the information stored in the encrypted authorization tag in a manner that cannot be interfered with by the signal-processing software or the user.

(ii) In operation, the combination of hardware and platform software enforces the transmission limits imposed by the encrypted authorization tag.

(iii) The combination of hardware and platform software holds the radio in compliance despite corruption due to failures in the signal-processing software. If the hardware/platform combination cannot maintain compliance, it turns off the radio.

A change in the platform software that potentially affects its behavior in any of these three respects should require re-approval under the class III permissive change procedure.[12]

During tests, each combination of hardware and platform software should be exercised with programs that emulate a wide variety of possible signal-processing software packages, adequate to drive the hardware/platform combination to the limits of its operation.

(b) Signal-processing software

In testing each instance of signal-processing software, the applicant should demonstrate that the device complies with the technical rules in the service for which the software is intended. The applicable rules will usually cover such characteristics as operating frequencies, maximum power, and out-of-band emissions. In some services, the rules may also govern bandwidth and specific modulations.

The Commission might reserve to itself (and its TCBs) the authority to create an encrypted software authentication tag, to prevent any signal-processing software from executing on any SDR without prior Commission approval.

Signal-processing software need be tested on only a single platform.[13] The signal-processing software does not control hardware directly. Moreover, because all certified

---

[12] *See* Notice at para. 25.

[13] *Cf.* 47 C.F.R. Sec. 15.31(i) (certification of a device that provides for external peripherals requires only one test using representative peripherals, not all possible combinations).

hardware/platform software combinations will read and enforce software tags in the same way, tests of signal-processing software on a single platform are adequate to prevent unsafe operation on any certified platform.

CONCLUSION

Software defined radios will improve spectrum efficiency, speed the availability of new services, and give the Commission new flexibility in spectrum allocations. All of these benefits can be achieved more safely, and at less expense, if the Commission recognizes a distinction between platform software and signal-processing software, and permits applicants to test signal-processing software separately from the combination of hardware and platform software.

Respectfully submitted,

/s/

Mitchell Lazarus
FLETCHER, HEALD & HILDRETH, P.L.C.
1300 North 17th Street, 11th Floor
Arlington, VA 22209
703-812-0440
Counsel for Vanu, Inc.

March 19, 2001

APPENDIX A

Vanu was formed in 1998 to explore the feasibility of building software radios using object oriented computer languages running on general purpose processors. This approach to software radio was initially investigated by the founders of Vanu in the SpectrumWare Project at Massachusetts Institute of Technology, which began in 1995. Project participants recognized that the rapid rate of improvement in microprocessor speed would soon bring the implementation of complex signal processing software systems into the realm of software. We believe this paradigm shift in the implementation of wireless communications systems will enable more efficient spectrum use, interoperation between historically incompatible radio systems, and much faster acceptance and adoption of advances in digital communications.

Vanu continues to focus on object oriented software that is portable across multiple platforms, and that supports independent specification and download of software radio applications. But the extent to which Vanu uses software to implement signal processing distinguishes us from other radio developers. In the nomenclature of the SDR Forum, Vanu develops "software radios" as opposed to "software defined radios." Pushing the digitization closer to the antenna permits our products much greater flexibility to adapt the nature of the signal processing performed by the radio. Vanu is currently involved in commercial partnerships to develop software radio products and is participating in Step 2B of the armed services' JTRS program. We are also engaged in a cooperative agreement with the National Institute of Justice to develop a prototype software radio interoperability device targeted at law enforcement needs.

SERVICE LIST

Chairman Michael Powell
Federal Communications Commission
445 12th Street, S.W.
Washington, D.C. 20554

Commissioner Harold Furchtgott-Roth
Federal Communications Commission
445 12th Street, S.W.
Washington, D.C. 20554

Commissioner Susan Ness
Federal Communications Commission
445 12th Street, S.W.
Washington, D.C. 20554

Commissioner Gloria Tristani
Federal Communications Commission
445 12th Street, S.W.
Washington, D.C. 20554

Bruce France, Acting Chief
Office of Engineering and Technology
Federal Communications Commission
445 12th Street, S.W., Room 7C-155
Washington, D.C. 20554

Julius P. Knapp, Chief
Policy & Rules Division
Federal Communications Commission
445 12th Street, S.W., Room 7B-133
Washington, D.C. 20554

Karen E. Rackley, Chief
Technical Rules Branch
Federal Communications Commission
445 12th Street, S.W., Room 7A-161
Washington, D.C. 20554

John A. Reed
Senior Engineer
Technical Rules Branch
Office of Engineering and Technology
Federal communications Commission
445 12th Street, S.W., Room 7A-140
Washington, DC 20554

Hugh Van Tuyl
Technical Rules Branch
Office of Engineering and Technology
Federal communications Commission
445 12th Street, S.W.
Washington, DC 20554